



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

**APPEAL FROM THE EXAMINER TO THE BOARD
OF PATENT APPEALS AND INTERFERENCES**

In re Application of: Richard P. TARQUINI
Serial No.: 10/004,192
Filing Date: October 31, 2001
Group Art Unit: 2157
Examiner: Nano, Sargon N.
Title: SYSTEM AND METHOD FOR UNIFORM RESOURCE LOCATOR FILTERING
Docket No.: 10017555-1

MAIL STOP: APPEAL BRIEF PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Dear Sir:

AMENDED APPEAL BRIEF

Applicant has appealed to the Board of Patent Appeals and Interferences from the decision of the Examiner mailed July 13, 2005, finally rejecting Claims 1-20. Applicant filed a Notice of Appeal on August 26, 2005. Applicant respectfully submits herewith this Amended Appeal Brief.

REAL PARTY IN INTEREST

The present application was assigned to Hewlett-Packard Company as indicated by an assignment from the inventor recorded on March 19, 2002 in the Assignment Records of the United States Patent and Trademark Office at Reel 012723, Frame 0968. The present application was subsequently assigned to Hewlett-Packard Development Company, L.P. as indicated by an assignment from Hewlett-Packard Company recorded on September 30, 2003 in the Assignment Records of the United States Patent and Trademark Office at Reel 014061, Frame 0492.

RELATED APPEALS AND INTERFERENCES

There are no known appeals or interferences that will directly affect or be directly affected by or have a bearing on the Board's decision in this pending appeal.

STATUS OF CLAIMS

Claims 1-20 stand rejected pursuant to a Final Office Action mailed July 13, 2005. Claims 1-20 are presented for appeal.

STATUS OF AMENDMENTS

No amendment has been filed subsequent to the mailing of the Final Office Action.

SUMMARY OF CLAIMED SUBJECT MATTER

Embodiments of the present invention as defined by independent Claim 1 are directed toward a method for Uniform Resource Locator (URL) filtering comprising receiving an event notification upon the occurrence of an event associated with a received URL, searching, in response to the event notification, a lexical search tree data structure (100) storing a plurality of URLs for the received URL and processing the received URL in response to the received URL not matching any of the plurality of URLs stored in the lexical search tree data structure (100). (at least at page 3, lines 24-32; page 4, lines 1-5

and 15-32; page 5, lines 1-32; page 6, lines 1-32; page 7, lines 1-32; page 8, lines 1-22; and figures 1-4).

Embodiments of the present invention as defined by independent Claim 10 are directed toward a system for Uniform Resource Locator (URL) filtering comprising a web server (16) operable to receive a URL request from a client (12), and a filter (18) operable, upon receiving an event notification relating to the URL request from the web server (16), to search a lexical search tree data structure (100) storing a plurality of URLs for the received URL. The filter (18) is further operable to process the received URL in response to the received URL not matching any of the plurality of URLs. (at least at page 3, lines 24-32; page 4, lines 1-5 and 15-32; page 5, lines 1-32; page 6, lines 1-32; page 7, lines 1-32; page 8, lines 1-22; and figures 1-4).

Embodiments of the present invention as defined by independent Claim 17 are directed toward a method for Uniform Resource Locator (URL) filtering comprising: receiving an event notification from a web server (16) upon the occurrence of a URL map event; determining, in response to receiving the event notification, a hash value for a URL received by the web server (16) from a client (12); determining a branch (104, 110, 112) associated with a root node (102) of a lexical search tree data structure (100) corresponding to the hash value, the lexical search tree data structure (100) storing a plurality of URLs, the branch (104, 110, 112) along with the root node (102) representing at least one URL of the plurality of URLs, the branch (104, 110, 112) having one or more leaf nodes (106, 120, 122) linked hierarchically to one another, each leaf node (106, 120, 122) representing an element of the at least one URL; traversing only the branch (104, 110, 112) to find a match between the received URL and the at least one URL; and processing the received URL in response to the received URL not matching the at least one URL. (at least at page 3, lines 24-32; page 4, lines 1-5 and 15-32; page 5, lines 1-32; page 6, lines 1-32; page 7, lines 1-32; page 8, lines 1-22; and figures 1-4).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Claims 1, 10 and 11 are rejected under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 5,996,011 issued to Humes (hereinafter “*Humes*”).

2. Claims 2-9 and 12-20 are rejected under 35 U.S.C. §103(a) as being unpatentable over *Humes* in view of U.S. Patent No. 6,631,369 issued to Meyerzon (hereinafter “*Meyerzon*”).

ARGUMENT**A. Standard****1. 35 U.S.C. § 102**

Under 35 U.S.C. § 102, a claim is anticipated only if each and every element as set forth in the claim is found in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 2 U.S.P.Q.2d 1051 (Fed. Cir. 1987); M.P.E.P. § 2131. In addition, “[t]he identical invention must be shown in as complete detail as is contained in the . . . claims” and “[t]he elements must be arranged as required by the claim.” *Richardson v. Suzuki Motor Co.*, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989); *In re Bond*, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990); M.P.E.P. § 2131.

2. 35 U.S.C. § 103

To establish a *prima facie* case of obviousness under 35 U.S.C. § 103, three basic criteria must be met: First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings; second, there must be a reasonable expectation of success; and finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *In re Vaeck*, 947 F.2d 488, (Fed. Cir. 1991); M.P.E.P. § 2143. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant’s disclosure. *Id.* Further, the mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination. *In re Mills*, 916 F.2d 680 (Fed.

Cir. 1990); M.P.E.P. § 2143.01. Additionally, not only must there be a suggestion to combine the functional or operational aspects of the combined references, but also the prior art is required to suggest both the combination of elements and the structure resulting from the combination. *Stiftung v. Renishw PLC*, 945 F.2d 1173, 1183 (Fed. Cir. 1991). Moreover, where there is no apparent disadvantage present in a particular prior art reference, then generally there can be no motivation to combine the teaching of another reference with the particular prior art reference. *Winner Int'l Royalty Corp. v. Wang*, 202 F.3d 1340, 1349 (Fed. Cir. 2000).

B. Argument

1. First Ground of Rejection (Claims 1, 10 and 11)

Claims 1, 10 and 11 were rejected under 35 U.S.C. §102(e) as being anticipated by *Humes*. Of these claims, Claims 1 and 10 are independent. Applicant respectfully submits that independent Claims 1 and 10 are patentable over the cited *Humes* reference, and thus remaining Claim 11 which depends from 10 is also patentable.

Embodiments of the present invention generally involve a system and method for Uniform Resource Locator (URL) filtering that involves searching a signature set, such as a plurality of URLs, for a target signature, such as a target URL, while reducing the number of comparisons (at least at page 3, lines 24-26). For example, in one embodiment of Applicant's invention, a lexical search tree data structure (100) is used to store the URLs to be searched in a structured and organized way (at least at page 3, lines 27-30; page 6, lines 4-32; page 7, lines 1-32; page 8, lines 1-22; and figures 3 and 4). In some embodiments of Applicant's invention, the lexical search tree data structure (100) comprises root nodes (102), branches (104, 110, 112), and one or more leaf nodes (106, 120, 122) linked hierarchically to one another (at least at page 6, lines 4-32; page 7, lines 1-32; page 8, lines 1-22; and figures 3 and 4). In some embodiments of Applicant's invention, a branch (104, 110, 112) along with its corresponding root node (102) and leaf nodes (106, 120, 122) represents one or more URLs (at least at page 6, lines 4-32; page 7, lines 1-32; page 8, lines 1-22; and figures 3 and 4). Accordingly, independent Claim 1,

for example, recites “receiving an event notification upon the occurrence of an event associated with a received URL,” “searching, in response to said event notification, a lexical search tree data structure storing a plurality of URLs for said received URL” and “processing said received URL in response to said received URL not matching any of said plurality of URLs stored in said lexical search tree data structure” (emphasis added).

In the Final Office Action, the Examiner appears to refer to column 3, lines 10-13, of *Humes* as purportedly disclosing searching a “lexical search tree data structure” as recited by Claim 1 (“see col. 3 lines 10-13, *Humes* discloses comparing the requested URL to an ‘allow list’” (Final Office Action, page 2)). Applicant respectfully disagrees. *Humes* appears to disclose that a requested URL is compared to an “allow list” which contains URLs of web pages which have been approved for display to the user and, if the requested URL is not on the “allow list,” comparing the requested URL to a “deny list” (*Humes*, column 3, lines 10-22). Applicant respectfully submits that neither the “allow list” nor the “deny list” of *Humes* is equivalent to a “lexical search tree data structure” as recited by Claim 1. To the contrary, *Humes* recites:

The method of the present invention first compares the requested URL to an “allow list” which contains URLs of web pages which have been approved for display to the user. If the requested URL is found in the allow list, the entire associated web page is, accordingly, forwarded to the user for downloading or viewing. If, however, the requested URL is not found in the allow list, the requested URL is then compared to a “deny list,” (or “forbidden list”) which functions in much the same manner as that of the prior art systems. If the requested URL is found in the forbidden list, a message is transmitted to the user’s computer indicating that access to the web page is forbidden (hereinafter referred to as a “FORBIDDEN” page).

(*Humes*, column 3, lines 10-22). Thus, *Humes* appears to disclose that the requested URL is, or can be, compared to every URL on such “allow list” and/or “deny list” of *Humes* to locate a match corresponding to the requested URL. In fact, the *Humes* system of URL comparison is at least one problem Applicant’s invention is directed toward

solving (“allows easy identification of the root node in the lexical search tree which corresponds to the first character in the target signature . . . [t]hus, all the root nodes do not have to be searched” (page 13 of Applicant’s specification, lines 16-21)). Accordingly, for at least these reasons, Applicant respectfully submits that *Humes* does not anticipate independent Claim 1.

Independent Claim 10 recites “a web server operable to receive a URL request from a client” and “a filter operable, upon receiving an event notification relating to said URL request from said web server, to search a lexical search tree data structure storing a plurality of URLs for said received URL, said filter further operable to process said received URL in response to said received URL not matching any of said plurality of URLs” (emphasis added). As discussed above, *Humes* does not disclose or even suggest searching “a lexical search tree data structure” as recited by Claim 10. Thus, at least for the reasons discussed above in connection with independent Claim 1, *Humes* also does not anticipate independent Claim 10.

Therefore, independent Claims 1 and 10, and Claim 11 which depends from independent Claim 10, are allowable.

2. Second Ground of Rejection (Claims 2-9 and 12-16)

Claims 2-9 and 12-16 were rejected under 35 U.S.C. §103(a) as being unpatentable over *Humes* in view of *Meyerzon*. Claims 2-9 and 12-16 depend respectively from independent Claims 1 and 10. As discussed above, independent Claims 1 and 10 are patentable over the cited reference. Therefore, Claims 2-9 and 12-16 that depend respectively therefrom are also patentable. Moreover, *Meyerzon*, which the Examiner applies combination with *Humes* to reject Claims 2-9 and 12-16, does not remedy, nor did the Examiner rely on *Meyerzon* to remedy, at least the deficiencies of *Humes* discussed above in connection with independent Claims 1 and 10. Accordingly, Claims 2-9 and 12-16 are clearly patentable over the cited references.

3. Second Ground of Rejection (Claims 17-20)

Claims 17-20 are rejected under 35 U.S.C. §103(a) as being unpatentable over *Humes* in view of *Meyerzon*. Of these claims, Claim 17 is independent. Applicant respectfully submits that independent claim 17 is patentable over the cited references, and thus remaining Claims 18-20 which depend therefrom are also patentable.

Independent Claim 17 recites “receiving an event notification from a web server upon the occurrence of a URL map event,” “determining, in response to receiving said event notification, a hash value for a URL received by said web server from a client,” “determining a branch associated with a root node of a lexical search tree data structure corresponding to said hash value, said lexical search tree data structure storing a plurality of URLs, said branch along with said root node representing at least one URL of said plurality of URLs, said branch having one or more leaf nodes linked hierarchically to one another, each leaf node representing an element of said at least one URL,” “traversing only said branch to find a match between said received URL and said at least one URL” and “processing said received URL in response to said received URL not matching said at least one URL” (emphasis added).

As discussed above in connection with independent Claim 1, *Humes* appears to disclose that a requested URL is compared to an “allow list” which contains URLs of web pages which have been approved for display to the user and, if the requested URL is not on the “allow list,” comparing the requested URL to a “deny list” (*Humes*, column 3, lines 10-22). Applicant respectfully submits that neither the “allow list” nor the “deny list” of *Humes* is equivalent to a “lexical search tree data structure” as recited by Claim 17. Moreover, neither the “allow list” nor the “deny list” of *Humes* appears to disclose or even suggest “a branch,” “a root node” or a “leaf node” as recited by Claim 17, nor has the Examiner explicitly identified any such disclosure in the *Humes* reference. Additionally, the *Humes* system appears to compare the URL to each entry on the “allow list” and, if not matched, against each entry on the “deny list” (*Humes*, column 3, lines 10-22). In contrast, independent Claim 17 recites “traversing only said branch to find a

match between said received URL and said at least one URL" (emphasis added). Accordingly, for at least these reasons, Claim 17 is patentable over the cited references.

Further, *Meyerzon* does not appear to disclose or even suggest "a lexical search tree data structure," "a branch," "a root node" or a "leaf node" as recited by Claim 17 (the Examiner appears to cite *Meyerzon* for its purported teaching of a hash value (Final Office Action, page 8)). Accordingly, *Meyerzon* does not remedy the deficiencies of *Humes*. In the Final Office Action, the Examiner also appears to refer to column 3, lines 50-67, of *Humes* as disclosing at least "a branch associated with a root node of a lexical search tree data structure," the "branch having one or more leaf nodes linked hierarchically to one another" and the "branch along with said root node representing at least one URL" as recited by Claim 17 (Final Office Action, page 8). Applicant respectfully disagrees. Column 3, lines 50-67, of *Humes* recite the following:

Advantageously, each word in the dictionary has a number of variables associated with it, such as: 1) a variable that indicates whether the word, if found, should be replaced with the innocuous filler (or a specific replacement filler word may be indicated); 2) a variable that indicates what category of objectionableness the word belongs to (i.e., pornography, intolerance, crime, job hunting, etc.); 3) a variable that indicates what language the word is a part of (i.e., English, French, Spanish, etc.); 4) a base score variable that indicates how objectionable the word is; and 5) a bonus score variable that indicates whether the word is more objectionable when used in combination with other objectionable words. In this advantageous embodiment, the method provides for filtering the body of the web page with the words in the dictionary. If a word in the web page matches, then that word will either be replaced or not replaced with the filler, as indicated by the variable.

Applicant respectfully submits that neither the portion of *Humes* referred to above by the Examiner, nor elsewhere in *Humes*, discloses, teaches or even suggests "a branch associated with a root node of a lexical search tree data structure," the "branch having one or more leaf nodes linked hierarchically to one another" and the "branch along with said root node representing at least one URL" as recited by Claim 17. In fact, the portion

of *Humes* referred to above by the Examiner does not appear to have any relation to a URL or storage thereof as the cited portion of *Humes* relied on by the Examiner appears to be directed toward filtering content appearing on the web page (“filtering the body of the web page with the words in the dictionary” (*Humes*, column 3, lines 63-65)). As discussed above, *Meyerzon* does not remedy the deficiencies of *Humes*. Accordingly, independent Claim 17 is clearly patentable over the cited references and, therefore, Applicant respectfully submits that Claim 17, and Claims 18-20 that depend therefrom, are in condition for allowance.

CONCLUSION

Applicant has demonstrated that the present invention as claimed is clearly distinguishable over the art cited of record. Therefore, Applicant respectfully requests the Board of Patent Appeals and Interferences to reverse the final rejection of the Examiner and instruct the Examiner to issue a notice of allowance of all claims.

Although no fee is believed due with this Amended Appeal Brief, the Commissioner is hereby authorized to charge any fees or credit any overpayments to Deposit Account No. 08-2025 of Hewlett-Packard Company.

Respectfully submitted,



James L. Baudino
Registration No. 43,486

Date: March 10, 2006

Correspondence To:

L. Joy Griebenow
Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400
Tel. (970) 898-3884

CLAIMS APPENDIX

1. A method for Uniform Resource Locator (URL) filtering, comprising:
 - receiving an event notification upon the occurrence of an event associated with a received URL;
 - searching, in response to said event notification, a lexical search tree data structure storing a plurality of URLs for said received URL; and
 - processing said received URL in response to said received URL not matching any of said plurality of URLs stored in said lexical search tree data structure.
2. The method of claim 1, wherein said searching said lexical search tree comprises:
 - determining a hash value for said received URL;
 - determining a branch associated with a root node of said lexical search tree data structure corresponding to said hash value, said branch along with said root node representing at least one URL of said plurality of URLs, said branch having one or more leaf nodes linked hierarchically to one another, each leaf node representing an element of said at least one URL; and
 - traversing only said branch to find a match between said at least one URL and said received URL.
3. The method of claim 2, wherein said determining a hash value comprises:
 - determining a first element of said received URL; and
 - determining a hash value for said first element.
4. The method of claim 3, said hash value being the ASCII code for said first element.
5. The method of claim 2, wherein said traversing only said branch comprises comparing successive elements of said received URL with successive elements of said at least one URL stored in successive leaf nodes of said one or more leaf nodes so long as said successive elements of said received URL match said successive elements of said at least one URL.

6. The method of claim 2, wherein said traversing only said branch further comprises:

determining a twig associated with said branch at a point of divergence between said at least one URL and said received URL, said twig representing a terminating substring of a second URL of said plurality of URLs; and

traversing said twig to find a match between a terminating substring of said received URL and said terminating substring represented by said twig.

7. The method of claim 6, wherein said traversing said twig comprises comparing successive elements of said terminating substring of said received URL with successive elements of said terminating substring of said second URL represented by said twig so long as said successive elements match.

8. The method of claim 5, wherein said traversing only said branch further comprises:

setting a current node pointer to point to a leaf node of said one or more leaf nodes;

setting a target signature pointer to point to an element of said received URL;

in response to a value of said leaf node pointed to by said current node pointer being equal to a wild card character and a value of the element pointed to by said target signature pointer being equal to a value of the next leaf node following the leaf node pointed to by said current node pointer, updating said current node pointer to point to a leaf node following said next leaf node.

9. The method of claim 1, wherein said receiving said event notification comprises receiving said event notification upon the occurrence of an event selected from the group consisting of a URL map event and a receive raw data event.

10. A system for Uniform Resource Locator (URL) filtering, comprising:
a web server operable to receive a URL request from a client; and
a filter operable, upon receiving an event notification relating to said URL request from said web server, to search a lexical search tree data structure storing a plurality of

URLs for said received URL, said filter further operable to process said received URL in response to said received URL not matching any of said plurality of URLs.

11. The system of claim 10, wherein said event notification relates to an event selected from the group consisting of a URL map event and a receive raw data event.

12. The system of claim 10, further comprising:

a computer readable medium with computer program logic recorded thereon for searching said lexical search tree data structure, said computer readable medium comprising:

means for determining a hash value for said received URL;

means for determining a branch associated with a root node of said lexical search tree data structure corresponding to said hash value, said branch along with said root node representing at least one URL of said plurality of URLs, said branch having one or more leaf nodes linked hierarchically to one another, each leaf node representing an element of said at least one URL; and

means for traversing only said branch to find a match between said at least one URL and said received URL.

13. The system of claim 12, wherein said means for determining a hash value comprises:

means for determining a first element of said received URL; and

means for determining a hash value for said first element.

14. The system of claim 13, wherein said means for traversing only said branch comprises means for comparing successive elements of said received URL with successive elements of said at least one URL stored in successive leaf nodes of said one or more leaf nodes so long as said successive elements of said received URL match said successive elements of said at least one URL.

15. The system of claim 14, wherein said means for traversing only said branch further comprises:

means for determining a twig associated with branch at a point of divergence between said at least one URL and said received URL, said twig representing a terminating substring of a second URL of said plurality of URLs; and

means for traversing said twig to find a match between a terminating substring of said received URL and said terminating substring represented by said twig.

16. The system of claim 15, wherein said means for traversing said twig comprises means for comparing successive elements of said terminating substring of said received URL with successive elements of said terminating substring of said second URL represented by said twig so long as said successive elements match.

17. A method for Uniform Resource Locator (URL) filtering, comprising:
receiving an event notification from a web server upon the occurrence of a URL map event;

determining, in response to receiving said event notification, a hash value for a URL received by said web server from a client;

determining a branch associated with a root node of a lexical search tree data structure corresponding to said hash value, said lexical search tree data structure storing a plurality of URLs, said branch along with said root node representing at least one URL of said plurality of URLs, said branch having one or more leaf nodes linked hierarchically to one another, each leaf node representing an element of said at least one URL;

traversing only said branch to find a match between said received URL and said at least one URL; and

processing said received URL in response to said received URL not matching said at least one URL.

18. The system of claim 17, said receiving said event notification comprising receiving a notification parameter from said web server, said notification parameter pointing to a data structure storing said received URL.

19. The system of claim 17, further comprising notifying said web server of a match between said received URL and said at least one URL.

20. The system of claim 17, further comprising:
registering with a web server to receive notification upon the occurrence of said URL map event.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None